



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/607,592	06/30/2000	Dayman Pang	042390.P8716	1180

7590

10/22/2003

Donna Jo Coningsby
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
7th Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025

EXAMINER

BASOM, BLAINE T

ART UNIT

PAPER NUMBER

2173

DATE MAILED: 10/22/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/607,592

Applicant(s)

PANG, DAYMAN

Examiner

Blaine Basom

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 06 October 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☐ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 9.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

Art Unit: 2173

DETAILED ACTION

Response to Arguments

The Examiner acknowledges the Applicant's amendments to claims 1, 4-6, 13, and 15-17. Regarding claims 1-5 and 13-16, the Applicant argues that Chari (U.S. Patent No. 6,046,742) fails to teach linking an associated graphical component with a console user interface (CUI) running CUI code and configuration kernel code under a graphical user interface, in order to configure a remote device according to a device configuration command, as has been added to independent claims 1 and 13. The Examiner respectfully disagrees with this argument. As is shown below, Chari presents an "MIB Manager module," which is considered a console user interface, and an "MIB," which is considered a configuration kernel, and wherein this MIB Manager Module and MIB are linked to a graphical user interface in order to configure a remote device according to a plurality of possible configuration commands. The MIB Manager Module "runs" MIB Manager Module code and MIB code, by calling modules which receive and change values in the MIB (for example, see column 8, lines 16-27). Because the MIB manager module and MIB are not directly accessed by the user - the MIB Manager module and MIB are only accessed through a graphical user interface (for example, see column 14, line 42 – column 15, line 5), the MIB Manager module is considered to run the MIB Manager module code and MIB code "under" the graphical user interface. Thus Chari in fact teaches linking an associated graphical component with a console user interface (CUI) running CUI code and configuration kernel code under a graphical user interface, in order to configure a remote device according to a device configuration command.

Art Unit: 2173

Regarding claims 6-10, the Applicant argues that Takimoto (U.S. Patent No. 6,041,350) does not disclose a configuration kernal (CK) having code to configure a device from a configuration, a console user interface (CUI) having code to update the configuration, and a graphical user interface (GUI) having code to receive an update to the configuration in response to a user action, wherein the CUI runs the CUI code and the CK code under the GUI, as has been added to independent claim 6. The Examiner respectfully disagrees with this argument. As is shown below, Chari presents an "MIB" and "SBEC" which are used to configure a device according to a plurality of configuration commands; the MIB is considered a configuration kernal having code to configure a remote device according a configuration, and this SBEC is considered a console user interface having code to update the configuration. The SBEC "runs" SBEC code and MIB code in order to simulate the behavior of a command to configure a remote device (for example, see column 5, lines 14-40). Because the SBEC and MIB are not directly accessed by the user – the SBEC and MIB are only accessed through a graphical user interface (for example, see column 6, lines 3-22), the SBEC is considered to run the SBEC code and MIB code "under" the graphical user interface. Thus Takimoto in fact discloses a CK having code to configure a device from a configuration, a CUI having code to update the configuration, and a GUI) having code to receive an update to the configuration in response to a user action, wherein the CUI runs the CUI code and the CK code under the GUI.

With respect to claims 17-21, the Applicant argues that the combination of Chari, note above, and Kekic et al. (U.S. Patent No. 5,999,179) fails to teach a console user interface that runs console user interface code and configuration kernal code under a graphical user interface, as has been added to independent claim 17. However, as is described above, Chari in fact

Art Unit: 2173

teaches such a console user interface running console user interface code and configuration kernel code under a graphical user interface. It is therefore understood that the combination of Chari and Kekic comprises this console user interface.

With respect to claims 11 and 12, the Applicant argues that the combination of Takimoto, note above, and the Applicant's admitted prior art fails to teach a console user interface that runs console user interface code and configuration kernel code under a graphical user interface. However, as is described above, Takimoto in fact teaches such a console user interface running console user interface code and configuration kernel code under a graphical user interface. It is therefore understood that the combination of Chari and the Applicants admitted prior art comprises this console user interface.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in-

(1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effect under this subsection of a national application published under section 122(b) only if the international application designating the United States was published under Article 21(2)(a) of such treaty in the English language; or

(2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that a patent shall not be deemed filed in the United States for the purposes of this subsection based on the filing of an international application filed under the treaty defined in section 351(a).

Claims 1-5 and 13-19 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,046,742, which is attributed to Chari. In general, Chari presents a method for the organized display of information for remote devices in a computer network. This display of

Art Unit: 2173

device information allows a user to easily view and alter the configuration information of a particular device. More particularly, this display of device information allows a user to view and alter the Management Information Base (MIB) of a remote device, wherein the MIB of a device comprises configuration information, which is used by the device to configure itself (see column 6, lines 19-32).

To begin, it is noted that the method disclosed by Chari is implemented on a computer (see column 6, lines 35-40). Therefore, and specifically regarding claim 13, it is understood that the method is implemented on a computer-readable medium comprising computer-executable instructions. As for both claims 1 and 13, the method of Chari involves presenting a graphical user interface to the user, whereby this graphical user interface includes forms which display the MIB values of a particular remote device. Moreover, these MIB values are displayed in dialog boxes (see column 13, lines 16-20). It is therefore understood that these dialog boxes are a graphical component of the forms of Chari, and that further, some form of graphical programming language is necessary to create such dialog boxes. Thus Chari teaches creating a graphical component, namely a dialog box, with a graphical programming language. More specifically, Chari discloses that the MIB values that are modifiable by the user are displayed in white dialog boxes (see column 14, lines 42-45). Therefore, such white dialog boxes, i.e. graphical components, are associated with device configuration commands. For example, Chari discloses an MIB variable, "coolingFanMinSpeed," which is modifiable through a white dialog box (see column 14, lines 45-48). Altering this variable value affects, i.e. configures, the system fans of the remote device (see column 14, line 57). The dialog box associated with "coolingFanMinSpeed" is therefore associated with a device configuration command for

Art Unit: 2173

configuring the system fans of the remote device. Consequently, Chari teaches associating graphical components, specifically dialog boxes, with device configuration commands.

Furthermore, Chari discloses that upon entering a new value for an MIB variable in a white dialog box, specific software components are then responsible for implementing the configuration change on the remote device. Particularly, an “MIB Manager Module” is implemented to adjust the MIB of the remote device, which in turn configures the remote device according to the device configuration command (see column 14, line 64 – column 15, line 5). Because the MIB Manager Module is also used to retrieve and display to the user data from a remote device, i.e. console (see column 8, lines 16-19), and because retrieving this data necessitates interfacing with the remote console, the MIB Manager Module is considered a “console user interface.” Similarly, because the MIB itself contains the basic data objects used to configure a remote device (see column 2, lines 40-50) the MIB is considered a “configuration kernel.” And finally, since the MIB Manager Module is implemented in response to the adjustment of a value in a white dialog box, wherein the MIB Manager Module specifically alters the data in the MIB in order to configure the remote device (see column 14, line 64 – column 15, line 5), the MIB Manager Module and the MIB are interpreted to be linked to that dialog box. It is therefore understood that Chari teaches linking an associated graphical component with a console user interface and configuration kernel, namely the MIB Manager Module and MIB, respectively, and wherein the console user interface and configuration kernel have code to configure a remote device according to the device configuration command. Because the MIB manager module and MIB are not directly accessed by the user, the MIB Manager module and MIB are only accessed through the graphical user interface, the MIB

Art Unit: 2173

Manager module is considered to run the MIB Manager module code and MIB code “under” the graphical user interface. Thus Chari more specifically teaches linking the associated graphical component with a console user interface running console user interface code and configuration kernel code under the GUI to configure a remote device according to a device configuration command. The MIB Manager Module further communicates updated MIB data, which is displayed in dialog boxes in the forms (see column 13, lines 16-23). Consequently, the forms disclosed by Chari reflect the state of the MIB as communicated by the MIB Manager Module. Chari thus teaches building a graphical user interface from linked graphical components, the console user interface, and the configuration kernel, in order to reflect a state of the configuration kernel as communicated by the console user interface.

As per claims 2 and 14, Chari teaches that associating a graphical component with a device configuration command can be performed using a macro. For example, as described above, the dialog box associated with “coolingFanMinSpeed” is associated with a device configuration command for configuring the system fans of the remote device. As additionally described by Chari, the MIB Manager Module is used to associate the dialog box with the “coolingFanMinSpeed” MIB variable (see column 13, lines 24-34). Such a module may include functions, which are considered equivalent to macros (see column 7, lines 50-60). Because of this, the MIB Manager Module is considered to incorporate a macro. Therefore, it is interpreted that a macro of the MIB Manager Module is used to associate the “coolingFanMinSpeed” dialog box with a device configuration command.

Regarding claim 3, the dialog box associated with “coolingFanMinSpeed” is associated with a device configuration command, wherein the value entered into the dialog box is used to

Art Unit: 2173

control the minimum speed below which a fan on the remote device is considered malfunctioning (see column 13, lines 24-27). This is considered equivalent to "adding a control to a dialog" as recited in claim 3.

In reference to claims 4, 5, 15, and 16, the graphical user interface disclosed by Chari is created using dialog boxes, and the MIB Manager Module and MIB to which the dialog boxes are linked, as explained above. According to Chari, these components may be implemented as object-oriented software components (see column 7, lines 50-60). Therefore, it is interpreted that building the graphical user interface disclosed by Chari requires compiling or interpreting dialog boxes, the MIB Manager Module code, and the MIB code on a general purpose computer system.

Claims 6-10 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,041,350, which is attributed to Takimoto. In general, Takimoto discloses a network management system that controls one or more network element management systems. A network element management system is a device which is used to control a network element. More specifically, a network element management system includes a management information database (MIB) and a behavior execution controller (BEC) (see column 7, line 66 – column 8, line 2). The MIB stores managed objects, which represent network element resources. It is interpreted that altering managed objects manages, i.e. configures, the resource (see column 1, lines 39-53). The BEC of the network element management system is used to manipulate the managed objects (see column 8, lines 5-16). Regarding the claimed invention, the network

Art Unit: 2173

management system of Takimoto is an apparatus used to configure a network element management system.

As per claim 6, the network management system disclosed by Takimoto includes an MIB and a simulated behavior execution controller (SBEC). The MIB of the network management system stores duplicates of the managed objects stored in the MIB of the network element management system (see column 5, lines 15-20). Modifying one of these managed objects in the network management system correspondingly modifies, or re-configures, the MIB of the network element management system (see column 5, lines 25-36). And because a managed object is created via object-oriented programming (see column 1, lines 33-34), it is interpreted that it is implemented through computer program code. Therefore, because it has code used to configure the network element management system, the MIB of the network management system disclosed by Takimoto is considered a "configuration kernel" like that recited in claim 6. The SBEC, which is understood to be implemented with computer code, is used to simulate the manipulation of the managed objects in the MIB of the network element management system (see column 8, lines 32-39). Because the result of this simulation is used to update the configuration of the MIB of the network element management system (see column 5, lines 25-36), and because it models the network element management system, i.e. console, the SBEC of the network management system is considered a "console user interface" as recited in claim 6. In addition, the network management system disclosed by Takimoto is interpreted to include a graphical user interface having code to receive an update to the configuration of the network element management system in response to a user action (see column 5, line 66 - column 6, line 11). As the MIB and SBEC are not directly accessed by the user - the MIB and SBEC are only

Art Unit: 2173

accessed through the graphical user interface, the SBEC is understood to run SBEC and MIB code “under” the graphical user interface. Lastly, it is understood that the network element management system disclosed by Takimoto includes a communications mechanism for communicating the received update from the graphical user interface to the SBEC, for communicating the updated configuration from the SBEC to the MIB, and for communicating the device configuration from the MIB to SBEC, and from the SBEC to the graphical user interface, in order to reflect a state of the MIB as communicated by the SBEC. Specifically, a “user interface controller,” “scenario controller,” and “transaction controller,” communicate the received update from the graphical user interface to the SBEC (see column 6, lines 3-11). The SBEC directly interacts with the MIB of the network management system (see column 6, lines 11-17), so it is interpreted that there must be some communication mechanism for communicating the updated configuration from the SBEC to the MIB, and for communicating the device configuration from the MIB to the SBEC. The device configuration is communicated from the SBEC to the graphical user interface using the “user interface controller” and “transaction controller” (see column 6, lines 11-22).

In reference to claim 7, the MIB of the network management system disclosed by Takimoto includes code to configure a device. As described above, this code is used to implement managed objects, which are stored in the MIB. Since these managed objects are realized as objects via object-oriented programming (see column 1, lines 33-34), and by the well-known definition of an “object,” it is interpreted that the managed objects comprise variables and functions (as an additional motivation for this interpretation, column 1, lines 35 – 39 shows that managed objects include variables, and column 3, lines 63-65 shows that managed objects

Art Unit: 2173

comprise functions). Moreover, as shown in figure 1, the MIB of the network management system includes managed objects, referred to in the drawing as “DMO_{xx}”, which are linked together in a tree or graph. Therefore, it is interpreted that the MIB also includes a data structure.

In regard to claim 9, the SBEC of the network management system disclosed by Takimoto includes code to update the configuration of a network element management system. As described above, the code of the SBEC is used to configure the network element management system by simulating the manipulation of the managed objects in the MIB of the network element management system. Such manipulations involve commands to create, delete, read, set, and execute managed objects or their attributes (see column 1, lines 39-53). Therefore it is interpreted that the code of the SBEC includes commands from this command set (as an additional motivation for this interpretation, column 10, lines 18-28 shows that the SBEC uses the command to read attributes of a managed object).

Regarding claims 8 and 10 Takimoto et al. discloses that the code of the SBEC is used to configure the network element management system by simulating the manipulation of the managed objects in the MIB of the network element management system. This simulation involves manipulating the duplicate managed objects stored in the MIB of the network management system (see column 8, lines 32-39). Such manipulations involve a library of commands to create, delete, read, set, and execute managed objects or their attributes (see column 1, lines 39-53). Takimoto further shows that a “scenario controller” and a “transaction controller” interpret requests from the graphical user interface and deliver commands from this library of commands to the SBEC to manipulate the managed objects in the MIB (see column 10,

lines 7-29). Therefore, these commands are also indirectly linked to the graphical user interface. Consequently, since the MIB, SBEC, and graphical user interface all linked to these commands, the code to configure the network element management system, i.e. the MIB of the network management system, is considered to reside in a library linked to the SBEC and graphical user interface. The code for updating the configuration of the network element management system, i.e. the SBEC of the network management system, is similarly considered to reside in a library linked to the SBEC and the graphical user interface.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 17-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over the U.S. Patent of Chari, which is described above, and also over U.S. Patent No. 5,999,179, which is attributed to Kekic et al. (and hereafter referred to as "Kekic"). As shown above, Chari discloses a method and computer-readable medium for presenting a graphical device management application. This graphical device management application is used to configure and maintain a remote device. Chari specifically discloses that the graphical user interface of this application comprises a plurality of graphical components, such as dialog boxes. As described above, these dialog boxes are linked to an MIB Manager Module, which in turn is responsible for retrieving

Art Unit: 2173

and setting MIB variables in order to configure the remote device. The MIB Manager Module and MIB are respectively analogous to the console user interface and configuration kernel recited in the present application. In addition, the MIB Manager module runs MIB Manager module code and MIB code under the graphical user interface, as is described above. As further described above, it is interpreted that creating such a graphical device management application, as disclosed by Chari, necessitates building its graphical user interface from the linked graphical components, the MIB Manager Module, and the MIB, in order to reflect a state of the MIB as communicated by the MIB Manager Module. Chari, however, does not explicitly disclose how the graphical user interface is built. In other words, Chari does not teach: interrogating the MIB Manager Module for a list of configuration commands that described the state of the MIB; comparing a configuration command to a register of commands that identifies an associated graphical component for each configuration command, wherein the configuration command describes the state of the MIB for the remote device, and the registered command identifies a graphical component associated with the configuration commands; identifying the registered command that matches the configuration command; and initializing, as a result of identifying the match, the graphical component to a corresponding state of the MIB, as is expressed in each of claims 20 and 21.

Similar to the Patent of Chari, the U.S. Patent of Kekic concerns the provision of a graphical user interface for configuring and maintaining a remote device (see column 6, lines 14-25). Kekic discloses that this graphical user interface includes various graphical components, which like the dialog boxes presented by Chari, are used to display and modify MIB data (see column 31, lines 37-42). Regarding the claimed invention, Kekic further discloses a “visual

Art Unit: 2173

element manager builder,” which is used to build such graphical user interfaces, referred to as “element managers” (see column 28, lines 10-46). This visual element manager builder comprises a panel entitled “Select the MIB Files to Use with the Manager,” with which a user chooses an MIB for which the element manager will be built to display and modify (see column 30, line 47 – column 31, line 15). Also included in the visual element manager builder is a “hotspot definition panel,” with which users create one or more “hotspots,” which like the dialog boxes of Chari, are the graphical components that display MIB values and allow them to be modified (see column 31, lines 16-42). Lastly, Kekic discloses that the visual element manager builder comprises a panel entitled “Select the Variable(s) with Each Hotspot,” which is used to associate each hotspot with an MIB variable (see column 34, lines 3-25). It is understood that this association between the hotspot and MIB variables is based on the type of MIB variable and the type of hotspot (see column 34, lines 26-31), or in other words, on a comparison between the type of MIB variable and hotspot. Consequently, when associating hotspots with MIB variables, the user compares each hotspot with one or more MIB variables, and conversely, compares each MIB variable with one or more hotspots. Furthermore, it is understood that as a result of the association, the graphical component is initialized to display the value of the MIB variable. And since an MIB variable is modified to configure the remote device, an MIB variable is considered to represent a configuration command. Similarly, each hotspot represents a command, as a hotspot is used to modify an MIB variable. Thus Kekic teaches: interrogating a file for a list of configuration commands, specifically MIB variables, that describe the state of an MIB; comparing such a configuration command to a register of commands, or more specifically hotspots, which identify an associated graphical component for each configuration command;

Art Unit: 2173

identifying the registered command that matches the configuration command; and initializing, as a result of identifying a match, the graphical component to a corresponding state of the MIB.

It would have therefore been obvious to one of ordinary skill in the art, having the teachings of Chari and Kekic before him at the time the invention was made, to modify the graphical user interface taught by Chari such that it may be built using the method taught by Kekic. In other words, and because the MIB Manager Module is responsible for retrieving MIB variables as described above, it would have been obvious to build the graphical user interface of Chari by: interrogating the MIB Manager Module for a list of configuration commands that described the state of the MIB; comparing a configuration command to a register of commands that identifies an associated graphical component for each configuration command, wherein the configuration command describes the state of the MIB for the remote device, and the registered command identifies a graphical component associated with the configuration commands; identifying the registered command that matches the configuration command; and initializing, as a result of identifying the match, the graphical component to a corresponding state of the MIB, as is taught by Kekic. It would have been advantageous to one of ordinary skill to utilize such a combination because the method presented by Kekic for creating a graphical user interface to configure and maintain a remote device provides an easier way to create such an interface, without writing computer code (see column 28, lines 32-46).

Referring to claim 17, the method disclosed by Chari teaches the idea of initializing a graphical component associated with a configuration command to a corresponding state of a remote device's configuration kernel. For example, Chari explicitly shows that a dialog box,

Art Unit: 2173

which is associated with a device configuration command for configuring the system fans of the remote device, is initialized with the “coolingFanMinSpeed” value. This “coolingFanMinSpeed” value is obtained from the management information base (MIB) of the remote device and represents the minimum speed below which a fan on the remote device is considered malfunctioning (see column 13, lines 24-27). The “coolingFanMinSpeed” value therefore represents a state of the MIB for a remote device (see column 13, lines 24-34). Because the MIB defines the aspects and components of the remote device, i.e. the configuration of the device (see column 2, lines 41-50), the MIB of the remote device is considered a “configuration kernel,” like that expressed in claim 17. As shown by reference number 1718 in figure 17, the initialized graphical component, or more specifically, the dialog box, is displayed on a window of a remote workstation. Moreover, Chari presents the idea of receiving an update to the configuration command from a user action on the associated graphical component. Continuing with the “coolingFanMinSpeed” example, this is done by typing in a new value into the “coolingFanMinSpeed” dialog box associated with the command (see column 14, lines 49-55). In response to receiving a new value for the dialog box, an “MIB Manager Module” disclosed by Chari is used to check the value to determine if it is within a pre-defined range (see column 14, lines 56-63). Therefore, it is inherent that the updated “coolingFanMinSpeed” value, which represents a configuration command, is sent to the MIB Manager Module. And because the MIB Manager Module is also used to retrieve and display to the user data from the remote device, i.e. console (see column 8, lines 16-19), the MIB Manager Module is considered a “console user interface,” as recited in claim 17. As described above in the rejection for claim 1, this console user interface runs console user interface code and the configuration kernel code under a

Art Unit: 2173

graphical user interface. Lastly, Chari notes that the MIB Manager Module updates the state of the MIB, i.e. configuration kernel, with the passed updated value, which represents a configuration command (see column 14, lines 64-66). Thus Chari teaches initializing a graphical component to a corresponding state of a configuration kernel; displaying on a window of a remote workstation this initialized graphical component; receiving an update to the configuration command from a user action on the associated graphical component; passing the updated configuration command to a console user interface that runs console user interface code and configuration kernel code under the graphical user interface; and updating by the console user interface the state of the configuration kernel with the passed updated configuration command. However, Chari does not explicitly disclose identifying a registered command that matches a configuration command, wherein as recited in claim 17, the configuration command describes a state of the configuration kernel for the networked device, and the registered command identifies a graphical component associated with the configuration command. Kekic, on the other hand, discloses this, as is shown above in the rejection for claim 20. Consequently, it is understood that the above-described combination of Chari and Kekic presents a method like that of claim 17, which is for configuring a networked device using a workstation.

As per claim 18, the idea of determining whether an updated configuration command is interdependent with a second configuration command, and refreshing the graphical component associated with the configuration command to reflect the updated state of the configuration kernel, i.e. MIB, is encompassed by the teachings of Chari. According to Chari, all displayed MIB variables that are dynamic are updated every five seconds (see column 13, line 65- column 14, line 1). Consequently, within 5 seconds of a user updating a dialog box value representing a

Art Unit: 2173

configuration command, all other displayed MIB values, including those that are interdependent with the updated value, are also updated. Therefore, because both ideas result in the adjustment of graphical components associated with interdependent configuration commands, the idea of updating dynamic MIB values every 5 seconds, as disclosed by Chari, is considered to encompass the idea of determining whether an updated configuration command is interdependent with a second configuration command, and refreshing the graphical component associated with the configuration command to reflect the updated state of the MIB.

As per claim 19, Chari notes that the MIB Manager Module updates the state of the MIB, i.e. configuration kernel, which is interpreted to be in the remote device (see column 14, lines 64-66). In other words, the MIB Manager Module uploads an updated state of the MIB to the MIB of the remote device.

Claims 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over the U.S. Patent of Takimoto, as described above, and also admitted prior art. As shown above, the network management system disclosed by Takimoto encompasses the apparatus of claim 6. More specifically, the network management system includes a management information database (MIB) and a simulated behavior execution controller (SBEC). The MIB includes code, i.e. managed objects, to configure a network element management system, as is described above. Moreover, these managed objects of the MIB of the network management system are duplicates of the managed objects that are stored in the MIB of the network element management system (see column 8, lines 29-32). Therefore, it is interpreted that the managed objects in the MIB of the network management system have been originally coded for operation in the MIB of a

Art Unit: 2173

network element management system. Like the MIB, the SBEC of the network management system includes code for updating the configuration of a network element management system, as is described above. More specifically, the SBEC is used to simulate the manipulation of the managed objects in the MIB of the network element management system. This simulation involves manipulating the duplicate managed objects stored in the MIB of the network management system in the same way that the network element management system manipulates the managed objects in its MIB (see column 8, lines 32-39). The network element management system includes a behavior execution controller (BEC) which is used to manipulate the managed objects (see column 8, lines 5-16). Therefore it is interpreted that the SBEC of the network management system directly corresponds, i.e. has the same functionality, of the BEC of the network element management system. Since the SBEC has the same functionality of the BEC, it is determined that the SBEC is equivalent to the BEC, which has been originally coded for operation on the network element management system. Thus the code for updating the network element management system, and the code for updating the configuration of a network element management system are reusable. However, Takimoto does not disclose that the code to configure a network element management system, i.e. the MIB of the network management system, and the code to update the configuration of a network element management system, i.e. the SBEC of the network management system, are firmware, as is expressed in claims 11 and 12. The Applicant, on the other hand, discloses admitted prior art, which conveys that configuration functionality can be implemented in firmware. Specifically, the Applicant states that "...the GUI developer ends up having to maintain command set aware source code that duplicates the functions of the CUI firmware implemented in the remote device's serial console." (see page 2,

Art Unit: 2173

lines 9-12) The benefits of firmware are well known in the art. Therefore, it would have been obvious to one of ordinary skill in the art, having the teachings of Takimoto and the admitted prior art, to modify the network management system of Takimoto, such that the MIB and the SBEC of the network management system are implemented in firmware. It would have been advantageous to one of ordinary skill to utilize such a combination because a firmware implementation is generally faster than a software implementation and requires less circuitry than a hardware implementation, as is well known in the art.

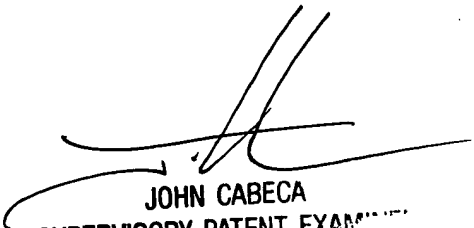
Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (703) 305-7694. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

btb


JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 21